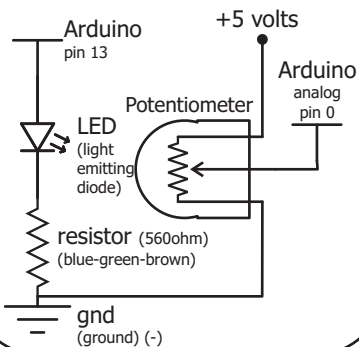
**WHAT WE'RE DOING:**

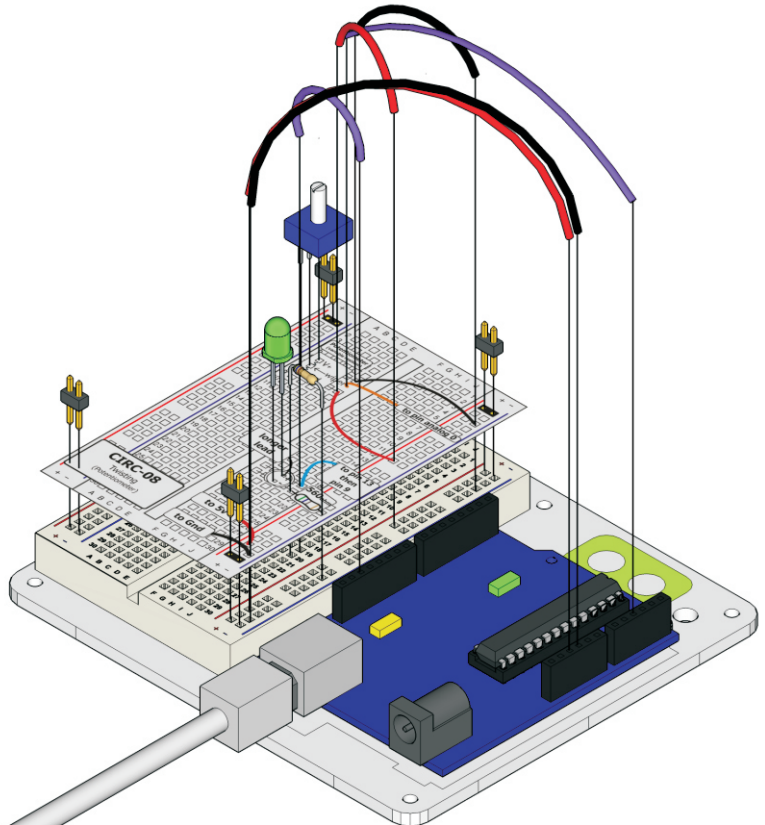
Along with the digital pins, the Arduino also has 6 pins which can be used for analog input. These inputs take a voltage (from 0 to 5 volts) and convert it to a digital number between 0 (0 volts) and 1024 (5 volts) (10 bits of resolution). A very useful device that exploits these inputs is a potentiometer (also called a variable resistor). When it is connected with 5 volts across its outer pins the middle pin will read some value between 0 and 5 volts dependent on the angle to which it is turned (ie. 2.5 volts in the middle). We can then use the returned values as a variable in our program.

THE CIRCUIT:**Parts:****CIRC-08
Breadboard Sheet
x1****2 Pin Header
x4****Potentiometer
10k ohm
x1****Wire****Green LED
x1****560 Ohm Resistor
Green-Blue-Brown
x1****Schematic****The Internet****..download..**

breadboard layout sheet

<http://ardx.org/BBS08>**..view..**

assembly video

<http://ardx.org/VIDE08>

CODE (no need to type everything in just click)**File > Examples > 3.Analog > AnalogInput**

(example from the great arduino.cc site, check it out for other great ideas)

```

/* Analog Input
 * Demonstrates analog input by reading an analog sensor on analog
 * pin 0 and turning on and off a light emitting diode(LED) connected to
digital pin 13.
 * The amount of time the LED will be on and off depends on the value obtained by
 * analogRead().
 * Created by David Cuartielles
 * Modified 16 Jun 2009
 * By Tom Igoe
 * http://arduino.cc/en/Tutorial/AnalogInput
 */

int sensorPin = 0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT); //declare the ledPin as an OUTPUT:
}

void loop() {
  sensorValue = analogRead(sensorPin); // read the value from the sensor:
  digitalWrite(ledPin, HIGH);          // turn the ledPin on
  delay(sensorValue);                  // stop the program for <sensorValue> milliseconds:
  digitalWrite(ledPin, LOW);           // turn the ledPin off:
  delay(sensorValue);                  // stop the program for for <sensorValue> milliseconds:
}

```

NOT WORKING? (3 things to try)**Sporadically Working**

This is most likely due to a slightly dodgy connection with the potentiometer's pins. This can usually be conquered by taping the potentiometer down.

Not Working

Make sure you haven't accidentally connected the potentiometer's wiper to digital pin 2 rather than analog pin 2. (the row of pins beneath the power pins)

Still Backward

You can try operating the circuit upside down. Sometimes this helps.

MAKING IT BETTER**Threshold switching:**

Sometimes you will want to switch an output when a value exceeds a certain threshold. To do this with a potentiometer change the loop() code to.

```

void loop() {
  int threshold = 512;
  if(analogRead(sensorPin) > threshold){
    digitalWrite(ledPin, HIGH);}
  else{ digitalWrite(ledPin, LOW);}
}

```

This will cause the LED to turn on when the value is above 512 (about halfway), you can adjust the sensitivity by changing the threshold value.

Fading:

Let's control the brightness of an LED directly from the potentiometer. To do this we need to first change the pin the LED is connected to. Move the wire from pin 13 to pin 9 and change one line in the code.

```
int ledPin = 13; ----> int ledPin = 9;
```

Then change the loop code to.

```

void loop() {
  int value = analogRead(potPin) / 4;
  analogWrite(ledPin, value);
}

```

Upload the code and watch as your LED fades in relation to your potentiometer spinning. (Note: the reason we divide the value by 4 is the analogRead() function returns a value from 0 to 1024 (10 bits), and analogWrite() takes a value from 0 to 255 (8 bits))

Controlling a servo:

This is a really neat example and brings a couple of circuits together. Wire up the servo like you did in CIRC-04, then open the example program Knob (**File > Examples > Servo > Knob**), then change one line of code.

```
int potpin = 0; ----> int potpin = 2;
```

Upload to your Arduino and then watch as the servo shaft turns as you turn the potentiometer.

MORE, MORE, MORE:

More details, where to buy more parts, where to ask more questions:

<http://ardx.org/CIRC08>